BASIC PROGRAMMING

Anleitung mit Programmier-Beispielen
Für ATARI VIDEO COMPUTER SYSTEM
Modell CX 2600 - VCS 800

EINLEITUNG

Die ATARI ProgrammCassette BASIC PROGRAMMING ist ein erster Schritt in die faszinierende Welt der Computer. Sie wird jedem Interessierten die Grundlagen der Computer-Programmierung zeigen.
BASIC ist die Abkürzung für die einfachste der höheren Programmiersprachen (Beginners All-purpose Symbolic Instruction Code). Sie läßt sich sehr leicht erlernen und stellt dennoch ein leistungsfähiges Werkzeug für jeden Programmierer dar.

Ein Computer-Programm besteht - einfach ausgedrückt - aus einer Folge von Anweisungen. Das Programm kontrolliert den Informationsfluß im Computer selbst.

Die ProgrammCassette BASIC PROGRAMMING kann - im Zusammenspiel mit dem VIDEO COMPUTER SYSTEM von ATARI - einfache Aufgaben ausführen. Natürlich ist die Zahl der Speicherplätze im Vergleich zu "professionellen" Computern beschränkt. Doch als Lerninstrument reicht es allemal.

WICHTIG
VOR JEDEM CASSETTEN-WECHSEL SCHALTER 1 (POWER) IN AUS- (OFF) STELLUNG BRINGEN.



Sollte das Bild auf Ihrem Fernsehgerät "durchlaufen" bitte über Justage "vertikaler Bildfang" nachregeln. Einzelheiten dazu bitte der Bedienungsanleitung Ihres Fernsehgerätes entnehmen.

Alle weiteren grundsätzlichen Bedienungs-Hinweise finden Sie in der Bedienungsanleitung für das ATARI VIDEO COMPUTER SYSTEM.

TASTENFELD

BASIC PROGRAMMING wird mit dem Paar Tastenfelder eingesetzt. Rechtes und linkes Tastenfeld werden mit den Anschlußbuchsen am Steuergerät verbunden. Dann beide Tastenfelder seitlich über Nut und Feder zusammenfügen. Daraus ergibt sich das Eingabefeld mti 24 Tasten - wichtig um das Programm einzugeben bzw. um die Ergebnis-Darstellung auf dem Bildschirm zu kontrollieren.

Doch vorher bitte noch die beiden Programmier-Schablonen (liegt der ProgrammCassetten-Verpackung als Zubehör bei) auf linkes bzw. rechtes Tastenfeld (LEFT CONTROLLER - RIGHT CONTROLLER - entsprechend den Anschlußbuchsen am Steuergerät) fixieren.

BILDSCHIRM-INFORMATION

Der Bildschirm ist in 6 Informations-Felder aufgeteilt.

1. PROGRAM-Feld
Zeigt die Anweisungen, die Sie dem Computer eingeben.

2. STACK-Feld
Zeigt die momentanen Ergebnisse Ihres Programmes beim Abarbeiten durch den Computer.

3. VARIABLES-Feld
Zeigt den Wert jeder Variablen während der Programm-Ausführung.

4. OUTPUT-Feld
Zeigt das jeweilige Endergebnis nach der Programm-Ausführung durch den Computer.

5. STATUS-Feld
Zeigt die Speicherkapazität zu jedem Zeitpunkt der Programm-Ausführung. Dieses Feld zeigt ebenfalls die Geschwindigkeit der Programm-Ausführung an.

6. GRAPHICS-Feld
Zeigt zwei Farbfelder, die über die Programmkontrolle bewegt werden können.

Bevor Sie nun mit dem Programmieren starten, bitte den Schalter 3 am Steuergerät in die b-Position bringen. Damit werden die Informations-Felder auf dem Bildschirm dargestellt. Wenn Sie nun den Schalter 3 in die a-Position bringen, verlöschen die Bezeichnungen der Informations-Felder auf dem Bildschirm und es erscheint nur noch das GRAPHICS-Feld.
Schalter 4 am Steuergerät ist bei BASIC PROGRAMMING ohne Funktion.

DER CURSOR

Der Cursor ist nichts anderes als ein kleines Rechteck auf dem Bildschirm und wird zur Programm-Eingabe benutzt.
Schalter 3 in b-Position bringen. Dann das Steuergerät aus- und wieder einschalten. Der Cursor befindet sich jetzt im PROGRAM-Feld.
Als nächste suchen Sie bitte die SHIFT-CONTROL-Taste (Mitte der unteren Tastenreihe auf linkem Tastenfeld). Diese Taste nun viermal drücken - die Farbe des Cursors ändert sich nun von weiß in rot, blau und grün. Wichtig: diese vier Farben entsprechen den jeweiligen Farbangaben für Anweisungen bzw. Eingaben auf den Tasten des Eingabe-Feldes.
Ein weißer Cursor gibt Anweisungen, alle anderen Cursor-Farben stehen für die Programm-Eingabe von Symbolen.

Bewegen des Cursors in den Informations-Feldern
Schalter 3 in b-Position bringen. Dann Steuergerät aus- und wieder einschalten.

Taste FORWARD drücken - der Cursor bewegt sich vom PROGRAM-Feld in das STACK-Feld. Taste FORWARD erneut drücken - der Cursor geht jetzt in das VARIABLES-Feld und bei weiterem Drücken in das OUTPUT-Feld.
Über BACKWARD-Taste kann der Cursor in das PROGRAM-Feld zurückbewegt werden. Die FORWARD- und BACKWARD-Tasten können für jeden Cursor-Schritt einzeln oder auch dauernd gedrückt werden.
Nun Schalter 3 in die b-Position bringen. Die Bezeichnungen der Information-Felder verlöschen und es erscheint ein Teil des GRAPHICS-Feldes. Nun den Cursor wie oben beschrieben durch die einzelnen Informations-Felder bewegen - Sie werden schnell feststellen, daß der Cursor nicht in das GRAPHICS-Feld gebracht werden kann.

Löschen der Informations-Felder
Schalter 3 in b-Position bringen. Dann Steuergerät aus- und wieder einschalten.
Links auf dem Eingabe-Feld finden Sie nun diverse Tasten (Farbe weiß) mit Anweisungen für die Informationsfelder: STATUS, PROGRAM, STACK, VARIABLES, OUTPUT und GRAPHICS. Wenn Sie jetzt die STATUS-Taste drücken, verlöscht das STATUS-Feld auf dem Bildschirm und das PROGRAM-Feld rückt in die erste Bildschirm-Zeile. Bitte forsetzen mit den Tasten PROGRAM, STACK, VARIABLES und OUTPUT. Jetzt erscheint das komplette GRAPHICS-Feld auf dem Bildschirm. Dann die GRAPHICS-Taste drücken - auf dem Bildschirm ist nun keines der Informations-Felder sichtbar. Nun wieder die STACK-Taste drücken - und das STACK-Feld befindet sich auf dem Bildschirm. Jetzt STACK-Feld löschen und das OUTPUT- und VARIABLES-Feld auf den Bildschirm bringen.
Mit dieser Übung haben Sie nun gelernt, daß Sie über den Schalter 3 in a- oder b-Position jedes Informations-Feld auf den Bildschirm bringen bzw. auf diesem löschen können.

Wichtig: Die Darstellung der Informations-Felder auf dem Bildschirm bzw. deren Löschen beeinflußt in keiner Weise die Programm-Ausführung.

PROGRAMM-AUSFÜHRUNG

Zu Beginn ein ganz einfaches Programm. Schalter 3 in b-Position bringen und Steuergerät aus- und wieder einschalten.
Es gibt übrigens noch einen anderen Weg, den Computer in diese Null-Position zu bringen:
Schalter 5 drücken. Sämtliche Werte und das Programm werden gelöscht.
Sollen nur die Zahlenwerte gelöscht und das Programm in die Start-Position gebracht werden, bitte den Schalter 6 drücken. Das Programm wird so nicht gelöscht sonder bleibt erhalten.

Jetzt die Felder STACK, VARIABLES, OUTPUT und GRAPHICS auf dem Bildschirm löschen.
Der weiße Cursor befindet sich nun rechts der Zahl 1 im PROGRAM-Feld. Cursor nun auf blau bringen und die Taste A drücken. Neben der 1 erscheint der Buchstabe A auf dem Bildschirm. (Jede Zeile ist numeriert - so kann leicht festgestellt werden, wann eine Zeile endet und eine neue Zeile beginnt).

Den Cursor nun auf rot bringen und die   Taste drücken. Neben dem A erscheint nun ein kleiner Pfeil. Jetzt Cursor auf blau und den Buchstaben B eingeben. Dann Cursor rot und + sowie 2 eingeben. Dann Cursor wieder auf weiß und NEW LINE eingeben. Wichtig: Für die Eingabe einer neuen Zeile muß der Cursor immer die Farbe weiß besitzen.

Und das zeigt jetzt der Bildschirm:

```
STATUS
SYMBOLS=57
SPEED=30
PROGRAM
1A←B+2
2
```

Sollte Ihnen bei der Programm-Eingabe ein Fehler unterlaufen, können Sie diesen mit der ERASE-Taste löschen. Die Taste kann unabhängig von der Farbstellung der SHIFT-Taste gedrückt werden.

Dazu eine kleine Übung - ausgehend von Ihrer letzten Programm-Eingabe. ERASE-Taste drücken. Cursor geht jetzt aus Zeile 2 neben die letzte Eingabe in der Zeile 1 (rechts neben die 2). ERASE-Taste erneut drücken und die 2 wird im Programm gelöscht. Jetzt Cursor auf rot und eine 1 eingeben. Danach Cursor auf weiß, BACKWARD-Taste drücken und den Cursor soweit nach links bewegen, bis er sich direkt neben dem B im Programm befindet. Erneut ERASE-Taste drücken und B wird im Programm gelöscht. Nun Cursor auf blau und A eingeben. Dann Cursor auf weiß und diesen mit der FORWARD-Taste bis zum Ende der Zeile 1 bewegen. Danach NEW LINE eingeben.

Der Cursor befindet sich immer direkt rechts vom zu löschenden Symbol. Hier der aktuelle Bildschirm:

```
STATUS
SYMBOLS=57
SPEED=30
PROGRAM
1A← A+1
2
```

Der Cursor befindet sich nun rechts der Programmzeilen-Angabe 2. Cursor auf grün und PRINT eingeben. Dann Cursor auf blau und A und NEW LINE eingeben. Hier der aktuelle Bildschirm:

```
STATUS
SYMBOLS=52
SPEED=30
PROGRAM
1A← A+1
2 PRINT A
3
```

Jetzt Cursor auf grün und GOTO eingeben. Dann Cursor auf rot und 1 eingeben. Das zeigt jetzt der Bildschirm:

```
STATUS
SYMBOLS=52
SPEED=30
PROGRAM
1A← A+1
2PRINT A
3GOTO 1
```

Das STATUS-Feld zeigt an, daß noch 52 "bytes" bzw. Symbole an Speicherkapazität zur Verfügung stehen. SPEED ist auf 30 fixiert. Jetzt Cursor auf weiß und die Taste SLOWER drücken. Jedes Drücken dieser Taste verringert die Geschwindigkeit. Das Drücken der FASTER-Taste vergrößert die Geschwindigkeit. Geschwindigkeit in diesem Zusammenhang bedeutet das Tempo, mit welchem der Computer die eingegebenen Programme abarbeitet.

Vor Programm-Start geben Sie jetzt SPEED=1 ein. Dann die STACK-Taste drücken. Danach zweimal RUN/HALT-Taste drücken und das Programm läuft. Sie können jetzt jeden Schritt mit dem der Computer das Programm abarbeitet beobachten.

```
STATUS
SYMBOLS=41
SPEED=1
PROGRAM
1A←A+1
2PRINT A
3 GOTO 1
STACK
A←3
```

Das Programm wird nun über die Taste RUN/HALT gestoppt. Demgegenüber werden über den Schalter 6 am Steuergerät (game reset) alle Werte im Programm gelöscht und unter Beibehaltung des Programm-Ablaufes dieses auf die Start-position gebracht.

Lassen Sie uns nun das Programm Schritt für Schritt analysieren. SPEED auf 60 bringen. Cursor auf weiß und STEP-Taste drücken. Damit wird das Programm über weiteres Drücken Schritt für Schritt abgearbeitet.

In der Zeile 1 zeigt das Programm A← A+1. Der Computer liest dies als A "wird" A+1. Achten Sie bitte auf das STACK-Feld. In Zeile 2 haben Sie dem Computer die Anweisung gegeben PRINT A. Der Computer soll also den Wert A (festgelegt in Zeile 1) im OUTPUT-Feld ausdrucken.

Wie das funktioniert sehen wir später. Zeile 3 sagt dem Computer GOTO 1. Der Computer geht also zur Zeile 1 zurück und findet dort einen neuen Wert für A. Das passiert nun bei jedem Programmdurchlauf - d.h. Finden eines neuen Wertes A, Ausdrucken dieses Wertes und Rückkehr zu Zeile 1.
Das geht solange, bis die zur Verfügung stehende Speicherkapazität aufgebraucht ist. Information darüber finden Sie unter SYMBOLS im STATUS-Feld.

Bringen Sie jetzt bitte das VARIABLES-Feld auf den Bildschirm. Dann SPEED auf 1 und löschen Sie sämtliche Werte im Programm über den Schalter 6 (game reset). Drücken Sie nun die RUN/HALT-Taste und beobachten Sie den Computer beim Abarbeiten des Programmes. Der Computer zeigt den momentanen Wert von A bei jedem weiteren Programm-Schritt. Und das ist die Information auf dem Bildschirm:

```
STATUS
SYMBOLS=27
SPEED=1
PROGRAM
1A←A+1
2PRINT A
3GOTO 1
STACK
PRINT 11
VARIABLES
A IS 11
```

Stoppen Sie nun das Programm und setzen alle Werte auf 0 (Schalter 6 - game reset). Dann das OUTPUT-Feld auf den Bildschirm bringen und SPEED auf 60 stellen. Hier der Bildschirm:

```
STATUS
SYMBOLS=52
SPEED 60
PROGRAM
1A←A+1
2 PRINT A
3 GOTO 1
STACK
OUTPUT
```

Nun mit der RUN/HALT-Taste das Programm starten. In Zeile 2 wird der Computer angewiesen, den Wert für A auszudrucken (PRINT A). Wird diese Anweisung im Programm erreicht, druckt der Computer den momentanen Wert von A im OUTPUT-Feld aus. Beobachten Sie bitte Symbols im STATUS-Feld.
Obwohl 1 im OUTPUT-Feld auf dem Bildschirm erscheint, zeigt der Computer den sich ändernden Wert von A. Dazu bitte Schalter 3 in a-Position. Hier der Bildschirm:

```
SYMBOLS=2
SPEED=60
1A←A+1
2PRINT A
3GOTO 1
PRINT 12
1
2
3
4
```

Wie können wir nun noch mehr Informationen im OUTPUT-Feld darstellen?

Dazu bitte STATUS-, PROGRAM- und STACK-Feld auf dem Bildschirm löschen. Hier der Bildschirm:

```
1
2
3
4
.
.
11
```

Und wie vermeiden wir das Überladen von Informationen im OUTPUT-Feld? Bitte das Programm stoppen und über Schalter 6 sämtliche Werte löschen. OUTPUT-Feld auf dem Bildschirm löschen und das PROGRAM-Feld zurückbringen. Dann über die FORWARD-Taste den Cursor an das Ende von Zeile 1 bewegen. Das ist der Bildschirm:

```
1 A←A+1
2 PRINT A
3 GOTO 1
```

Jetzt die Taste NEW LINE drücken. Wir wollen eine neue Anweisung einfügen. Dazu Cursor auf <u>grün</u> und CLEAR eingeben. Hier der Bildschirm:

```
1 A←A+1
2 CLEAR
3 PRINT A
4 GOTO 1
```

Nun die Felder STACK, VARIABLES und OUTPUT auf den Bildschirm bringen. Schalter 3 in a-Position. Programm starten. Wenn nun im Programm die Zeile 2 CLEAR abgearbeitet wird, erlöscht der Wert von A im OUTPUT-Feld und es wird der nächste Wert von A aus Zeile 3 gedruckt.

BASIC PROGRAMMING arbeitet maximal mit 2-stelligen Zahlen. Wenn also 99 erreicht wird, geht es automatisch zurück auf den Wert 0 für A.

MUSIK-PROGRAMMIERUNG

Jedesmal, wenn über die Taste NOTE (Cursor=rot) im Programm eine Zahl eingespeichert wird, ertönt beim folgenden Programmablauf im Lautsprecher des Fernseher ein entsprechender Ton.

Im Folgenden nun einige Demonstrations-Programme. Dazu bereiten Sie wieder Ihr Steuergerät durch Drücken des Schalter 6 vor - d.h. alle alten Programme werden gelöscht.
Hier die erste Eingabe:

```
1 NOTE ←NOTE + 1
2 GOTO 1
```

Jetzt wird das Programm gestartet! Bitte niedrige Geschwindigkeit wählen und im STACK-Feld das Abarbeiten des Programmes beobachten. Der jeweils aktuelle Wert von NOTE wird im VARIABLES-Feld angezeigt.

Dann das Programm stoppen und die Zeile 2 neu eingeben (Cursor=weiß, Cursor an das Ende von Zeile 1 und Taste NEW LINE drücken. Die alte Zeile 2 wird nun Zeile 3). Hier die Eingabe:

```
2 IF NOTE>6 THEN NOTE←0
```

Das ist der Bildschirm (STACK- und VARIABLES-Feld gelöscht):

```
1 NOTE←NOTE+1
2 IF NOTE>6 THEN NOTE
←0
3 GOTO 1
```

Mit den IF- und THEN-Anweisungen arbeitet das Programm so: Wenn sich etwas Bestimmtes ereignet, dann ist auch etwas Bestimmtes zu tun. Und so funktioniert das in unserem Programm: Wenn (IF) NOTE größer als 6, dann (THEN) wird NOTE auf 0 gesetzt.
Jetzt das Programm starten und im STACK-Feld beobachten. Wenn der Wert von NOTE nun 7 erreicht, wird dieser (>) größer als 6 und das Programm setzt ihn auf 0 zurück.

Jetzt ein anderes Programm für das gleiche Ergebnis, jedoch mit einem anderen Weg. Hier die Eingabe:

```
1 NOTE←C
2 C←C+1
3 IF C>7 THEN C←0
4 GOTO 1
```

Sobald das Programm läuft, sieht man, daß das gleiche Ergebnis wie vorher erreicht wird - nur die Noten sind stärker voneinander getrennt.
Nach Eingabe von PRINT C und CLEAR kann der jeweilige Wert von C im OUTPUT-Feld abgelesen werden. Die Eingabe eines Kommas (Cursor=grün) hinter dem ausgedruckten Wert und erneute Eingabe von PRINT C verhindern das Flimmern der Werte im OUTPUT-Feld.

Für die Variablen im Programm kann jeder Buchstabe des Alphabetes benutzt werden. Im folgenden Programm finden sich nun alle bisher angesprochenen Tasten-Funktionen wieder - und es wird nahezu die gesamte Speicher-Kapazität unseres Computer ausgenutzt. Nach Eingabe des Programmes bitte Bildschirmanzeige löschen, die Informationsfelder für STACK und OUTPUT auf den Bildschirm bringen und dann das folgende Programm starten:

```
1 D←7 CLEAR
2 NOTE←D PRINT D
3 D←D-1
4 IF D>7 THEN GOTO 1
5 IF D<8 THEN GOTO 2
```

In diesem Programm kommen zwei IF/THEN-Anweisungen zum Einsatz. Falls nun die Voraussetzungen der ersten Anweisung nicht eintreffen (Zeile 4) bearbeitet der Computer automatisch die nächste Zeile (Zeile 5). Bei genügend Speicherkapazität können mehrere Anweisungen zwischen den beiden IF/THEN-Anweisungen vorhanden sein. Übrigens - es können auch 2 Anweisungen in einer Zeile eingegeben werden (siehe Zeilen 1 und 2 im obigen Programm). Das spart für noch folgende Programm-Anweisungen Speicherkapazität.

KEY- UND PRINT-FUNKTION

Die KEY-Funktion (Cursor=rot) wird für die Eingabe einer Variablen in das laufende Programm benutzt. Das Programm slebst löscht dann KEY und ersetzt es mit einer Zahl, die Sie über das Tastenfeld eingeben. Falls keine Zahl eingegeben wird, interprtiert das Programm KEY als 0.
Hier ein einfaches Programm mit der KEY-Funktion:

```
SYMBOLS=50
SPEED=30
1 IF KEY>0 THEN NOTE←
KEY
2 GOTO 1
```

Jetzt das Programm starten und einige Tasten im rechten Tastenfeld drücken. Bei einiger Übung gelingt es Ihnen nun, eine kleine Melodie zu spielen.

Die KEY-Funktion wird außerdem noch zum Programmieren im GRAPHICS-Feld eingesetzt. Doch dazu später.

Nun eine Programm mit den KEY-, NOTE- und PRINT-Funktionen:

```
SYMBOLS=34
SPEED=30
1A ←KEY
2 NOTE ←A PRINT A
3 IF A=0 THEN CLEAR
4 GOTO 1

9
```

Im OUTPUT-Feld jetzt den Programm-Ablauf beobachten. Jede Zahleneingabe "spielt" nun einen entsprechenden Ton - und gleichzeitig wird diese Zahl im OUTPUT-Feld angezeigt.

Jetzt eine kleine Programmänderung. In Zeile 2 wird ein Komma hinter PRINT A eingefügt und dann das Programm erneut gestartet. Jetzt wird das Programm so viele Variablen in einer Zeile ausdrucken wie nur irgend möglich.

DIE PRINT-FUNKTION

Mit der PRINT-Funktion wird der Wert einer Variablen im OUTPUTfeld ausgedruckt. Darüber hinaus können jedoch auch komplette Worte über die PRINT-Funktion auf dem Bildschirm dargestellt werden. Worte müssen allerdings in Anfführungszeichen (") gesetzt werden. Bitte jetzt SPEED auf 8 setzen und dieses Programm eingeben:

```
1 PRINT "A"
2 PRINT "IST"
3 PRINT "A"
4 A←A+1
5 CLEAR
6 GOTO 1
```

Jetzt das Programm starten und das OUTPUT-Feld beobachten. Das Programm druckt nun die eingegebenen Worte aus - allerdings zeilenweise untereinander:

```
A
IST
12
```

Deshalb wird nun das Programm durch Eingabe eines Kommas am Ende von Zeile 1 und 2 abgeändert. Hier der neue Ausdruck:

```
A IST 15
```

Mit dem Komma in Zeile 1 wird das Programm angewiesen, den gesamten Inhalt von Zeile 2 im OUTPUT-Feld in der Anweisungszeile (d.h. Zeile 1) auszudrucken. Das Komma in Zeile 2 bewirkt, daß der Inhalt von Zeile 3 in Zeile 2 ausgedruckt wird. Diese Anweisungen können auch kürzer eingegeben werden:

```
1 PRINT "A","IS
T",A
2 A←A+1
3 CLEAR
4 GOTO 1

A IST 15
```

Eine weitere Kürzung dieses Programmes, wleche zudem noch Speicherkapazität einspart:

```
1 PRINT "A IST",A
2 A←A+1
3 CLEAR
4 GOTO 1

A IST 18
```

DAS GRAPHICS-FELD

Das blaue Rechteck auf dem Bildschirm stellt das GRAPHICS-Feld dar. Oben links befindet sich zudem noch ein kleines rotes Feld. Und darunter in gleicher Größe ein kleines weißes Feld. Mit einem speziellen Programm können diese beiden Felder unabhängig voneinander bewegt werden.
Dazu müssen jedoch erst einmal die jeweiligen Koordinaten festgelegt werden. Dem roten Feld rechnen wir jetzt die Zahl 1 zu. Diesem Feld entspricht die horinzontale Koordinate HOR 1 und die vertikale Koordinate VER 1 auf dem Tastenfeld. HOR 2 und VER 2 entsprechen dem zweiten, weißen Feld.

Zurück zum Ausgangspunkt - der linken, oberen Ecken im GRAPHICS-Feld. Für uns ist dies jetzt die 0-Position. Sind nun den Variablen HOR 1, VER 1, HOR 2 und VER 2 keine Werte zugeordnet (d.h. diese befinden sich auf 0) sind auch das kleine weiße und rote Feld übereinander in der 0-Position.
Wird nun einer Koordinaten einer größerer Wert als 0 zugeordnet (z.B. HOR 1←10), springt das jeweilige kleine Feld bei der Programmausführung auf die den Koordinaten entsprechende Position im GRAPHICS-Feld.

Bitte jetzt dieses Programm eingeben:

```
1 HOR 1←HOR 1+1
2 GOTO 1
HOR 1 IS 17
```

Das rote Feld bewegt sich nach Programmstart langsam nach rechts. Im VARIABLES-Feld sieht man den Wert von HOR 1 anwachsen - bis 99, der größten möglichen Zahl, um dann wieder auf 0 zurückzugehen. Bei HOR 1=99 hat das rote Feld die rechte Seite des Bildschirmes erreicht. Danach dann erneuter Start auf linker Seite.

Die horizontalen Koordinaten (HOR 1 und HOR 2) reichen also von 0 bis 99. Das gilt auch für die vertikalen Koordinaten (VER 1 und VER 2) - wobei 0 für den oberen und 99 für den unteren Rand des GRAPHICS-Feldes steht.

Dieses Programm eingeben:

```
1 HOR 1←HOR 1 +6
2 VER 1←VER 1 +8
3 HOR 2←HOR 2 +4
4 VER 2←VER 2 +4
5 IF HIT THEN NOTE←2
ELSE NOTE ←7
6 GOTO 1
```

Jetzt das Programm vom Bildschirm bringen und dafür sorgen, daß das GRAPHICS-Feld voll sichtbar ist. Geschwindigkeit auf 60 setzen und dann das Programm starten. Das Programm zeigt jetzt eine der Möglichkeiten, die beiden kleinen Felder auf dem Bildschirm zu bewegen. Es zeigt außerdem, wie die HIT- und ELSE-Funktionen eingesetzt werden.

In der Programmzeile 5 wird der Computer angewiesen, die Note mit dem Wert 2 erklingen zu lassen, wenn sich die beiden kleinen Felder berühren (IF HIT). HIT bedeutet also, daß die bei kleinen Felder die gleiche Koordinate besitzen.
Sonst (ELSE) wird der Computer die Note mit dem Wert 7 spielen. Das macht er auch - aber nur solange, wie sich die beiden kleinen Felder nicht berühren.
Wird nun in Zeile 5 ELSE NOTE←7 gelöscht, erklingt ausschließlich die Note mit dem Wert 2 und auch nur dann, wenn sich die beiden kleinen Felder berühren.

DIE MOD-FUNKTION

MOD ist eine arithmetische Operation, ähnlich der Division.
Bei BASIC PROGRAMMING kommt die sogenannte Integer-Division zum Einsatz. Das bedeutet, daß nur mit ganzen Zahlen gearbeitet werden kann und kein Rest zurückbleibt.
Beispiel: 14:5? Vielleicht 2 und 4/5? Oder 2 Rest 4? Beides ist richtig - nur bei BASIC PROGRAMMING ist das Ergebnis 2, da ja kein Rest angegeben wird.

Demnach gibt der Computer gleiche Antworten - egal ob 12:4 oder 13:4. Beide Ergebnisse lauten auf 3.

Nun zur MOD-Funktion. MOD zeigt den Rest bei Teilung zweier Zahlen. Daher ist 14MOD5 genau 4. Denn 14 geteilt durch 5 ist 2 Rest 4. MOD zeigt nur den jeweiligen Rest an - deshalb in diesem Beispiel für MOD die Zahl 4 - eben der Rest.

Und was bedeutet 13MOD4? Genau - 1, denn der Rest von 13 geteilt durch 4 beträgt 1. Und 12MOD4? Diesmal 0, denn 12 geteilt durch 4 geht glatt auf und es verbleibt keine Rest bzw. 0.

Mathematisch gesehen ist es sinnlos eine Zahl durch 0 zu teilen. In BASIC PROGRAMMING ergibt jede Teilung durch 0 (z.B. 5:0) wieder 0.

Prioritäten der mathematischen Operationen

BASIC PROGRAMMING arbeitet mit diesen Prioritäten:

1. x und : (höchste Priorität)
2. + und -
3. MOD
4. =
5. ← (geringste Priorität)

Gebrauch von Klammern

Der Einsatz von Klammern (Cursor=grün) gibt den angegebenen Zahlen die entsprechende Priorität innerhalb der Ausführung einer Rechenoperation.

Beispiel:

```
A 5+3x2 MOD7   Erster Schritt:    3x2=6
               Zweiter Schritt:   5+6=11
               Dritter Schritt:   11MOD7=4
A=4
```

Die gleiche Operation mit Klammern arbeitet wie folgt und gibt auch ein anderes Ergebnis:

```
A (5+3)x2MOD7  Erster Schritt:    5+3=8
               Zweiter Schritt:   8x2=16
               Dritter Schritt:   16MOD7=2
A=2
```

PROGRAMM-BEISPIELE

Die folgenden Programm-Beispiele zeigen die vielen Möglichkeiten, mit denen der Computer arbeiten kann. Achten Sie bitte auf die Bedeutung der entsprechenden Anweisungen (IF, THEN, HIT, ELSE, PRINT, KEY, u.s.w.).

Und wenn mal ein neues Programm nicht sofort arbeiten sollte, bitte dieses stoppen, es auf die START-Position zurückstellen und dann Schritt für Schritt über die STEP-Taste nach dem möglichen Fehler suchen.
Im STACK-Feld kann dann sehr gut gesehen werden, wie und warum der Computer so und nicht anders arbeitet.

Wir von Atari wünschen Ihnen jedenfalls viel Erfolg beim Ausprobieren der folgenden Programme und natürlich noch mehr Erfolg beim Entwickeln Ihrer eigenen Programme. Dazu ist BASIC PROGRAMMING ein erster, entscheidender Schritt.
Übrigens - wussten Sie schon, daß Atari auch Heimcomputer mit richtiger Schreibmaschinen-Tastatur, mit Drucker und Cassettenrecorder zur Programmspeicherung und dafür auch viele fertige Programmer herstellt?

ATARI ELEKTRONIK-Vertriebsgesellschaft mbH
Arndtstraße 16   2000 Hamburg 76

HIT

```
1 CLEAR PRINT HIT
2 HOR2←HOR2+1MOD
33
3 GOTO 1
```

MUSIK

```
1 NOTE←NOTE + KEY+
7
2 GOTO 1
```

```
1 A←KEY
2 IF A>0 THEN NOTE← A
- 1
3 GOTO 1
```

```
1 NOTE ←IF NOTE>4
THEN NOTE - 11 ELSE
NOTE + 1
2 NOTE←NOTE+2
3 NOTE←NOTE-2
4 GOTO 1
```

```
1 A←A+1
2 NOTE ←A
3 IF(A MOD2)=0 THEN
NOTE← 0 ELSE NOTE← 4
4 GOTO 1
```

UHREN-PROGRAMM

Bitte nur das OUTPUT-Feld beim Programmablauf auf den Bildschirm bringen. SPEED auf 30 oder 60 setzen.

```
1 CLEAR
2 A← A+1 MOD 60
3 IF A=0 THEN B← B+1
MOD 60
4 PRINT B,A
5 GOTO 1
```

PING-PONG-SPIEL

```
1 HOR 1← HOR 1 + KEY
2 HOR 2← HOR 2 + 8
3 VER 2← VER 2 - 3
4 IF HIT THEN VER 2← 99
, NOTE←7
5 GOTO 1
```

PONG-SPIEL (ohne Ton)

```
1 HOR 2← HOR 2 + KEY
2 IF VER 1>90 THEN
VER 1← 88
3 IF HIT THEN VER 1← 9
4 VER 1  VER 1 + IF
VER 1 MOD 2 THEN 8 ELSE 92
5 HOR 1← HOR 1+7
6 GOTO 1
```

ATARI CASSETTEN-PROGRAMM

Best.-Nr.	Deutsch	Englisch	Spiele	Regler
		ACTION		
CX 2601	PANZER/FLUGZEUGSPIEL	COMBAT	27	J
CX 2602	LUFT/SEESCHLACHT	AIR/SEABATTLE	27	J
CX 2604	WELTRAUM-KAMPF	SPACE WAR	17	J
CX 2605	SCHARFSCHÜTZEN	OUTLAW	16	J
CX 2613	ABENTEUER	ADVENTURE	3	J
CX 2622	FERNSEHFLIPPER	BREAKOUT	12	P
CX 2627	FLIEGENDER MENSCH	HUMAN CANNONBALL	8	J
CX 2630	CIRCUS	CIRCUS ATARI	8	P
CX 2631	SUPERMAN	SUPERMAN	1	J
CX 2632	SPACE INVADERS/ANGRIFF AUS DEM WELTRAUM	SPACE INVADERS	112	J
CX 2635	LABYRINTH	MAZE CRAZE	256	J
		SPORT		
CX 2606	HINDERNIS-RENNEN	SLOT RACERS	9	J
CX 2611	SUPER-AUTORENNEN (INCL. SPEZIALREGLER)	INDY 500	14	S
CX 2612	STRASSENRENNEN	STREET RACER	27	P
CX 2616	FUSSBALL	CHAMPION SHIP SOCCER	54	J
CX 2621	OLYMPISCHE SPIELE	VIDEO OLYMPICS	50	P
CX 2624	BASKETBALL	BASKETBALL	2	J
CX 2626	MINIGOLF	MINIATURE GOLF	2	J
CX 2625	BOWLING	BOWLING	6	J
CX 2629	FALLSCHIRM-SPRINGER	SKY DIVER	5	J
CX 2633	NACHT-FAHRER	NIGHT DRIVER	8	P
CX 2634	GOLF	GOLF	1	J
CX 2637	PUNKTE-RENNEN	DODGE 'EM	3	J
		SPIELE		
CX 2617	BACKGAMMON	BACKGAMMON	8	P
CX 2618	TIC-TAC-TOE	3 D-TIC-TAC-TOE	9	J
CX 2636	VIDEO-DAME	VIDEO-CHECKERS	19	J
CX 2641	BLOCKADE/ZEICHENSPIEL	SURROUND	14	J
CX 2642	MEMORY	CONCENTRATION/ HUNT & SCORE	8	K
CX 2643	GEHEIMCODE/NIM	CODEBREAKER	20	K
CX 2645	VIDEO-SCHACH	VIDEO CHESS	8	J
CX 2651	BLACK JACK/ 17 + 4	BLACK JACK	7	P
CX 2652	CASINO-SPIELE	CASINO	4	P
CX 2664	INTELLIGENZ-SPIELE	BRAIN GAMES	19	K
		LERNEN		
CX 2620	COMPUTER-TRAINING	BASIC PROGRAMMING	1	K
CX 2621	MATHEMATIK-TRAINING	FUN WITH NUMBERS/ BASIC MATH	8	J

Regler (Steuerknüppel in Grundausstattung)

				Best.-Nr.
J	= Steuerknüppel	Joystick	(2 Stck.)	CX 40-04
P	= Drehregler	Paddle Control	(2 Stck.)	CX 30-04
K	= Tastatur	Keyboard	(2 Stck.)	CX 50-04

Stand: Februar 1981